On Nearly Infallible Large Scale Visual Classification

Anonymous CVPR submission

Paper ID ****

Abstract

In this work we consider the problem of "nearly infallible classification".

1. Introduction

In this paper we study the problem of "nearly infallible large scale visual classification", creating visual classification systems can recognize as many as $10,000$ object categories yet almost never produces an incorrect answer.

How is it even remotely possible, given that the state of the art classification accuracy on 10,000 object categories is 16.7% [**?**]? Yet the answer is in the affirmative with a trivial solution. Object categories form a semantic hierarchy with many levels of abstraction. A "German Shepherd" is also a "dog", an "animal", a "living thing" and above all, an "entity". Therefore always predicting the root node "entity" would give an infallible classifier.

The answer is valid but utterly useless, for the obvious reason that we gain no information at all through the classification. We realize that we should ask our question in a more interesting way: *can we create a classification system that maximizes the information gain of its output yet maintains a very small error rate?*

Specifically, we can measure the information gain in the standard information theoretical sense, *i.e.* the decrease in uncertainty(entropy) of class probability distribution through classification. Without classification, our knowledge of the object in the input image is "entity", *i.e.* the probability distribution is uniform among the tens of thousands of leaf categories [1]. Given a classification output of "dog", our uncertainty decreases. While still uncertain of the specific breed of dog, we gain information by ruling out a large number of possibilities. Note that the question can be generalized to not just information gain, but any reward function that prefers more specific classification.

Our key observation is that we can allow the classification system to freely choose the level of abstraction and

[1] We may have a non-uniform prior and uncertainty is still well defined.

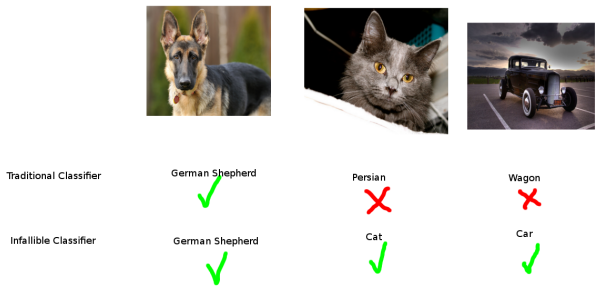

Figure 1. Conventional classifier versus infallible classifier.

therefore can ensure arbitrarily small error rate by selecting an appropriate level for each input. In other words, exploiting the semantic hierarchy, accuracy can be ensured by trading off information gain. For example, the classification system would output "dog" for an image of "German shepherd" if the confidence of "German shepherd" is insufficient to ensure a low error rate(Fig. 1).

Nearly infallible classification is practically important. In general, ensuring small error rate is always desirable because users prefer reliable and trustworthy classification systems. In many cases, erroneous information is worse than correct but less specific information, particularly when the information is actionable. For example, predicting "chanterelle" implies an edible type of mushroom, which if erroneous, may have serious consequences. "Mushroom" should be the preferred answer if near infallibility cannot be ensured.

The problem is of broad interest. The rationale extends beyond object classification to any visual tasks that involve high level semantic labels, *e.g.* detection, scene understanding, segmentation, describing images with sentences and so on. In this work we focus on basic multiclass object classification, which can serve as the building blocks of other tasks.

The problem is novel. Previous work in computer vision has been focused on certain particular levels of abstraction but not across the entire spectrum in a unified framework. Research in visual classification has been moving from basic levels, represented by a large body of work evaluated

on Caltech101/Caltech256/PASCAL, to sub-ordinate levels, represented by large scale classification and fine grained classification [?, ?, ?]. Most of the work does not consider the semantic hierarchy, using flat set of classes [?]. For those who do [?, ?], there is no notion of information gain or small error guarantee and the issue of automatically choosing the appropriate abstraction level is not systematically addressed. We will elaborate the differences in Sec. 2.

In this work, we study the nearly infallible classification problem and make the following contributions:

- To the best of our knowledge, we are the first to introduce the nearly infallible classification problem in computer vision, *i.e*. maximizing information gain under small error constraint given a semantic hierarchy.
- We solve the problem by a primal dual algorithm that's provably optimal under realistic conditions.
- We evaluate our method on large scale datasets up to 10,184 object categories. We demonstrate that our method significantly beats heuristic baselines and that we can build a nearly infallible system that's also practically useful.

2. Related Work

Multiclass classification. Multiclass classification, classifying the input into exactly one of multiple classes, is well studied in machine learning and extensively used in computer vision. Standard approaches include joint learning of a single model [?, ?] and reduction to binary problems, *e.g*. one-vs-one, one-vs-all, DAGSVM [?], ECOC [?]. These approaches assume a set of non-overlapping classes and 0-1 loss for misclassification. They do not apply to hierarchical classes in a straightforward manner.

Classification using hierarchy. There is a large body of work using hierarchy in image classification and annotation [17]. We divide the most relevant work into roughly three groups.

The first group of work uses the hierarchy to improve the performance of flat multiclass classification at the leaf nodes [3, ?, ?, ?, 1, ?]. The hierarchy is typically constructed automatically and does not necessarily align with the semantic hierarchy. This differs from our work where the problem involves a predefined semantic hierarchy and our classification system needs to output internal nodes as well as leaf nodes.

The second group of work uses hierarchy to define new performance measure different from the conventional 0-1 loss, typically penalizing less if the confused classes are closer on the hierarchy [5, ?, ?, ?, ?]. All these losses are defined based on where the path of the prediction diverge from the ground truth, the lower the better. In this case, a prediction on leaf node is always preferred, because given any prediction of internal node, one can always instead predict an arbitrary leaf node in the subtree with equal or less penalty. Our work differs in that to ensure a small error rate, it is necessary to prefer internal nodes in many cases.

The third group of work considers classification for both internal nodes and leaf nodes [21, 18, 13]. Hierarchy is used either to combine models[21], or to organize the classifiers [13, 18, ?], where multiclass or binary classifiers are learned for each node. Typically, a test example is classified at all semantic levels, *e.g*. by sending it down the hierarchy to a leaf node.

Our work is closely related to this line of work but differs in two significant ways. First, the issue of choosing the best abstraction level is our focus whereas it was only touched upon tangentially. For example, in [13], the model is shown to perform well for higher semantic levels when uncertain at lower levels, but it is unclear how to decide which semantic level to output. In [18], a reject threshold is introduced so that examples with low confidence will stay at internal nodes instead of going down the tree. However, it is unclear how to choose the threshold for each node and how to handle a hierarchy that's a directed acyclic graph(DAG) instead of a tree. Second, there was no notion of an overall performance measure or any error rate guarantee, whereas we provide a principled formulation and a theoretically rigorous treatment.

Cost sensitive classification. Cost sensitive classification extends flat multiclass classification by considering non-uniform misclassification loss. This also includes the aforementioned second group of work on classification using hierarchy. One approach is to calibrate the classifier outputs into probabilities and predict the class that minimizes the expected loss [?, 20]. Other approaches modify the surrogate loss function in classifier learning [14, 15]. These approaches do not consider overlapping classes. In this work, we extend the probability calibration approach to overlapping classes due to its general applicability to various classifiers, and use it as a sub-routine in our algorithm.

Classification with reject options. Classification with reject options extends binary classification by granting an option to abstain, for a particular cost [6, 2, 19]. The binary case is well understood, where the optimal classification rule is given by a threshold of the posterior probability [6] and the threshold is determined by the reject cost. Recent work has been focused on learning techniques that lead to such an optimal classifier [2, 19]. A generalization to multiclass, termed class selective rejection or nondeterministic classification, has been studied recently [11, 10, 12]. In the multiclass case, the classifier is allowed to output an *arbitrary* set of classes of its choice and the goal is to find the

optimal trade-off between misclassification loss, set size, error rate and reject rate. Our problem can be thought of as a special case where the classes are leaf nodes and admissible sets are internal nodes of the hierarchy. To the best of our knowledge, we are the first to draw the connection between class selective rejection and hierarchical visual classification. Our primal dual framework is similar to that in [**?**], but we establish more general conditions for strong duality with different proof techniques and also extends the results to non Bayesian optimal classifiers.

Multi-label annotation Multi-label annotation assigns multiple labels to an image[CITE something]. This is different from our problem in that those labels are typically for different objects present in the image rather than different levels of abstractions for the same object, and no special consideration is given for selecting the abstraction level. In our case, only one label of the chosen semantic level is predicted as it implies the higher level labels.

Learning with partial or incomplete labels Learning with partial or incomplete labels [7, 4] is concerned with weaker supervision, when a training example is only given a set of labels but only one of them is true(partial labels) or is give one label but the ground truth includes multiple labels(incomplete labels). The end tasks are still the conventional multiclass classification or multi-label annotation. In this work we assume that all examples are labeled to the leaf nodes and leave learning with partial labels(internal nodes) as future work.

Fine grained classification. Fine grained classification[CITE WORK] is a challenging problem focused on classification for sub-ordinate categories. Our framework can integrate fine grained classification techniques into a generic system by automatically selecting the proper classification level.

3. Problem setup

Let $H = (V, E)$ be a directed acyclic graph with a root at $\hat{v} \in V$. let $\mathcal{Y} \subset V$ be the set of leaf nodes. Let $\pi(y)$ be the set of nodes that are ancestors of node y *including* y itself. Let r_v be the reward for predicting a correct node v. One instance of the reward is information gain, but our framework works for any reward function. Assuming distribution of classes, the information gain is then $r_v = \log_2 |\mathcal{Y}| / \sum_{y \in \mathcal{Y}} [v \in \pi(y)]$ for node v. Let (X, Y) be a random example and its ground truth label drawn from a joint distribution $\mathbb{D}$ on $\mathcal{X} \times \mathcal{Y}$. Let $f : \mathcal{X} \to V$ maps an example $x \in \mathcal{X}$ to a node $v \in V$. Let $\epsilon > 0$ be the maximum error rate we tolerate, that is, we require $1 - \epsilon$ infallibility.

Let $R(f)$ be the (expected) reward of f.

$$R(f) = \mathbb{E} r_{f(X)}[f(Y) \in \pi(Y)]. \quad (1)$$

Let $\phi(f)$ be the (expected) infallibility of f, that is, the percentage of correct predictions.

$$\Phi(f) = \mathbb{E}[(f(X) \in \pi(Y)]. \quad (2)$$

Note that if we discard the hierarchy and keep only the leaf nodes, then $\pi(y) = \{y\}$. In this case, the infallibility is exactly the conventional multiclass classification accuracy with $0 - 1$ loss.

We can now define the optimization problem.

OP1. Nearly infallible classification

$$\begin{aligned} &\underset{f}{\text{maximize}} && R(f) \\ &\text{subject to} && \Phi(f) \geq 1 - \epsilon. \end{aligned}$$

We seek a classifier f that maximizes the reward with an infallibility of at least $1 - \epsilon$. One immediate observation is that OP1 always has a feasible solution $\hat{f}$, which maps all the examples x to the root node $\hat{v}$.

4. Theoretical Analysis

4.1. Unconstrained case

To start, we consider an easier problem OP1', maximizing the reward without the infallibility constraint.

OP2. Maximizing reward unconstrained.

$$\underset{f}{\text{maximize}} \qquad R(f) = \mathbb{E}\ r_{f(X)}[f(X) \in \pi(Y)]$$

We observe that we obtain the optimal solution if we have conditional probabilities $p_{Y|X}(y|x) = Pr(Y = y|X = x), y \in Y$ for any x. That is, given the example, we simply predict the node with the maximum expected reward using the conditional probabilities. If we have the conditional probabilities, the probabilities at the leaf nodes are simply the sum of the leaf node probabilities. We give details in the following lemma.

Lemma 4.1. *Let* $f^*(x) = \arg\max_{v \in V} r_v p_{Y|X}(v|x)$, *where* $p_{Y|X}(v|x) = \sum_{y \in Y} [v \in \pi(y)] p_{Y|X}(y|x)$. *Then* f^* *is the optimal solution to OP2.*

Proof. Observe that

$$\begin{aligned} R(f) &= \mathbb{E}_X \mathbb{E}_{Y|X}\ r_{f(X)}[f(X) \in \pi(Y)] \\ &= \mathbb{E}_X\ r_{f(X)} \sum_{y \in \mathcal{Y}} [f(X) \in \pi(y)] p_{Y|X}(y|X) \\ &= \mathbb{E}_X\ r_{f(X)} p_{Y|X}(f(X)|X) \\ &\leq \mathbb{E}_X\ \max_{v \in V} r_v p_{Y|X}(v|X) \\ &= \mathbb{E}_X\ r_{f^*(X)} p_{Y|X}(f^*(X)|X) = R(f^*). \end{aligned}$$

□

Note that it is not always possible to obtain the true conditional probabilities $p_{Y|X}(y|x)$, which means achieving the optimal bayesian error rate for flat classification. In fact, in many cases, the true conditional probabilities are deterministic, *i.e.*, belong to one of the leaf classes with probability 1. Then having true conditional probabilities are equivalent of knowning the ground truth labels. In this case it's meaningless to do infallible classification because everything is already infallible. However, for the moment, we assume this for convenience of developing the theory. We will relax this requirement later.

Before addressing the constraint in OP1, we first establish a sufficient condition of the reward setup so that infallibility is automatically guaranteed. That is, the optimal solution to OP2 is automatically nearly infallible.

Lemma 4.2. *Let $r_{max} = \max_{v\in V} r_v$ and f^* be the optimal solution to OP2. If $r_{\hat{v}}/r_{max} \geq 1-\epsilon$, where $\hat{v}$ is the root node, then $\Phi(f^*) \geq 1-\epsilon$.*

Proof. Assume to the contrary that $\Phi(f^*) < 1-\epsilon$. Then

$$\begin{aligned} R(f^*) =& \mathbb{E} r_{f^*((X)}[f^*(X) \in \pi(Y)] \leq \mathbb{E} r_{max}[f^*(X) \in \pi(Y)] \\ =& r_{max}\Phi(f^*) < r_{max}(1-\epsilon) = r_{\hat{v}} = R(\hat{f}), \end{aligned}$$

where $\hat{f}$ is the trivial solution that maps all examples to the root node. This contradicts that f^* is optimal. □

Lemma 4.2 shows that if the reward of the root node is relatively big, we automatically obtain infallibility. However, this is not generally the case, especially for the information gain setup. Therefore we have to address the constraint in OP1.

4.2. The constrained case

A general strategy to deal with constrained optimization is the Lagrange multipliers method. We first write the Lagrange function with the multiplier λ.

$$L(f,\lambda) = R(f) + \lambda(\Phi(f) - 1 + \epsilon) \tag{3}$$

Also we define the Lagrange dual function

$$J(\lambda) = \sup_f L(f,\lambda). \tag{4}$$

Given a λ, let f_λ be a function that achieves the supreme of $L(f,\lambda)$, that is,

$$J(\lambda) = L(f_\lambda, \lambda). \tag{5}$$

Note that in general f_λ may not be unique, but here we assume each λ maps to a unique f_λ generated by some deterministic procedure.

We then have the following standard results, first proved by Everett [9] in studying the multiplier method in resource allocation problems.

Lemma 4.3. *If there exists a $\lambda_\dagger \geq 0$ such that $\Phi(f_{\lambda^\dagger}) = 1-\epsilon$, then $f_{\lambda^\dagger}$ is an optimal solution to OP1 [9].*

Lemma 4.4. *If $0 \leq \lambda_1 < \lambda_2$, then $\Phi(f_{\lambda_1}) \leq \Phi(f_{\lambda_2})$ and $R(f_{\lambda_1}) \geq R(f_{\lambda_2})$ [9].*

Lemma 4.3 states that if we find a λ and its corresponding f_λ that maximizes the Lagrange $L(f,\lambda)$ such that the infallibility $\Phi(f)$ is exactly $1-\epsilon$, then f_λ is an optimal solution to OP1. In this case, strong duality is achieved. Lemma 4.4 states that the infallibility $\Phi(f_\lambda)$ is non-decreasing with λ. Note that these results are general. They apply regardless of how Φ and R are defined, without assuming any convexity or smoothness of $R(f)$ or $\Phi(f)$.

This suggests the following strategy for solving OP1. We first set $\lambda = 0$ and obtain f_0. Note this is equivalent to maximizing OP1 unconstrained, *i.e.* OP2. We then check if this solution satisfies the constraint *i.e.* $\Phi(f_0) \geq 1-\epsilon$. If so, f_0 is the optimal solution to OP1 and we are done. Otherwise, we have $\Phi(f_0) < 1-\epsilon$ and we do binary search to find $\lambda^\dagger > 0$ such that $f_{\lambda^\dagger} = 1-\epsilon$, provided that such $\lambda^\dagger > 0$ exists.

For this strategy to work, we need to address two questions. First, given any $\lambda \geq 0$, can we obtain f_λ, that is, maximizing the Lagrange $L(f,\lambda)$? Second, if $\Phi(f_0) < 1-\epsilon$, does a $\lambda^\dagger > 0$ exist such that $f_{\lambda^\dagger} = 1-\epsilon$, *i.e.* achieving strong duality? There are no general answers and for many problems the answers are no. We address them for our infallible classification problem in the following sections.

4.3. Maximizing Lagrange function

The answer to the first question is a simple yes. Plugging Eqn. 1 and Eqn. 2 into Eqn. 3 gives

$$L(f,\lambda) = \mathbb{E}(r_{f(X)} + \lambda)[f(X) \in \pi(Y)] + \lambda(\epsilon - 1).$$

Therefore obtaining f_λ is simply solving OP2 with a transformed reward $\tilde{r}_v = r_v + \lambda, \forall v \in V$ and Lemma 4.1 gives the Bayesian optimal solution, *i.e.*

$$f_\lambda(x) = \arg\max_{v\in V}(r_v + \lambda)p_{Y|X}(v|x). \tag{6}$$

4.4. Condition for strong duality

To address the second question(existence of $\lambda^\dagger$), we first observe that we can make λ large enough to make $\Phi(f_\lambda) \geq 1-\epsilon$.

Lemma 4.5. *Let $\bar{\lambda} = (r_{max}(1-\epsilon) - r_{\hat{v}})/\epsilon$, where $\hat{v}$ is the root node and $r_{max} = \max_{v\in V} r_v$. If $\Phi(f_0) < 1-\epsilon$, then $\bar{\lambda} > 0$ and $\Phi(f_{\bar{\lambda}}) \geq 1-\epsilon$.*

Proof. Note that $\bar{\lambda} > 0$ because otherwise Lemma 4.2 implies that $\Phi(f_0) \geq 1-\epsilon$. Let $\tilde{r}_v = r_v + \bar{\lambda}, \forall v \in V$ be the transformed rewards. It is easy to verify that $\tilde{r}_{\hat{v}}/\tilde{r}_{max} \geq 1-\epsilon$. Lemma 4.2 then implies that $\Phi(f_{\bar{\lambda}}) \geq 1-\epsilon$. □

Now we immediately have the following sufficient condition for the existence of $\lambda^\dagger$.

If $\Phi(f_0) < 1-\epsilon$ and $\Phi(f_\lambda)$ is continuous with respect to λ, then there exists a $\lambda^\dagger > 0$ such that $f_{\lambda^\dagger} = 1-\epsilon$.

Proof. By Lemma 4.5 there exists $\bar{\lambda} > 0$, $\Phi(f_{\bar{\lambda}}) \geq 1-\epsilon$. Intermediate value theorem then proves the conclusion. □

Lemma 4.6 states that it is sufficient for $\Phi(f_\lambda)$ to be continuous with respect to λ. Unfortunately it is not true in general and we can construct a simple example such that for certain $\epsilon > 0$, no such $\lambda^\dagger$ exists.

Consider the simplest hierarchy of two leaf nodes, class a and class b, and a root node class c. Let the rewards $r_a = r_b = 1$ and $r_c = 0$. Let $p_a(x) = P(y = a|x)$ be the conditional probability that an example x belongs to class a. Note that in this binary case, $p_a(x)$ complete determines the conditional multinomial distribution. It is easy to verify that for $0 \leq \lambda \leq 1$, $f_\lambda(x) = a$ if $p_a(x) \geq 0.5$ and $f_\lambda(x) = b$ otherwise. For $\lambda > 1$, $f_\lambda(x) = a$ when $p(x) \geq \frac{\lambda}{\lambda+1}$, $f_\lambda(x) = b$ when $p_a(x) \leq \frac{1}{\lambda+1}$ and $f_\lambda(x) = c$ otherwise.

Suppose $p_a(x)$ only takes two discrete values, $p_1 = 0.6$ and $p_2 = 0.4$. Let $\mu(p)$ be the percentage of examples such that $p(x) = p$. Here let $\mu(p_1) = \mu(p_2) = 0.5$, that is, half of examples have conditional probability of 0.6, another half 0.4. Then it is easy to verify that for $0 \leq \lambda < 1.5$, $\Phi(f_\lambda) = 0.6$, *i.e.* all examples are predicted to the leaf nodes. For $\lambda \geq 1.5$, $\Phi(f_\lambda) = 1$, *i.e.* all examples are predicted to the root node. Therefore there are only two possible values for $\Phi(f_\lambda)$ and it is not possible to achieve an arbitrary $1-\epsilon$.

In this artificial example, the discontinuity stems from the fact that the conditional class probability $p_{Y|X}(a|x)$ concentrates on only two values other than 0 and 1. Therefore when we vary λ, we have sudden jumps of $\Phi(\lambda)$. In practice, however, we expect the distribution of $p_{Y|X}(a|x)$ to be either *deterministic i.e.* $p_{Y|X}(a|x)$ is 1 or 0 for all x, or *continuous i.e.* has a density function over $[0,1]$, or a mixture of both *i.e.* for those that are not deterministic, they are continuous. In fact, for all these more realistic cases, $\Phi(\lambda)$ turns out to be continuous.

We first make various notions precise. Let $\Delta = \{q \in \mathbb{R}^{|\mathcal{Y}|-1} : q \succeq 0, \|q\|_1 \leq 1\}$ be the set of possible probability values of $|\mathcal{Y}|-1$ leaf nodes. Note that for $|\mathcal{Y}|$ leaf nodes there are only $|\mathcal{Y}|-1$ degrees of freedom. Let the last leaf node be y_1. With slightly abuse of notation, for $q \in \Delta$, we write q_y as the probability of any leaf node $y \in \mathcal{Y}$, with the understanding that $q_{y_1} = 1 - \|q\|_1$. We also use $q_v = \sum_{y \in \mathcal{Y}} [v \in \pi(y)] q_y$ to mean the probability of any node $v \in V$. Then it follows that if $q = \vec{p}_{Y|X}(x)$, then $q_v = p_{Y|X}(v|x), \forall v \in V$. Let $\Delta^\ddagger = \{q \in \Delta : \|q\|_\infty = 1 \vee q = 0\}$, *i.e.* the set of leaf probabilities for which one of the leaf nodes takes probability 1.

Let $\vec{p}_{Y|X} : \mathcal{X} \to \Delta$ be the function that maps an example to class probabilities of leaves. Let

$$Q = \vec{p}_{Y|X}(X) \tag{7}$$

be the class probabilities of X on leaves. Then Q is also a random variable. We call Q *deterministic* if $\Pr(Q \in \Delta^\ddagger) = 1$, *continuous* if Q has a probability density function p_Q with respect to Lebesgue measure on $\mathbb{R}^{|\mathcal{Y}|-1}$, and *partially deterministic but otherwise continuous* if $0 < \Pr(Q \in \Delta^\ddagger) < 1$ and there exists a conditional density function $p_{Q|Q \notin \Delta^\ddagger}$ of Q when $Q \notin \Delta^\ddagger$.

The case of deterministic Q is trivial because we know exactly the ground truth labels, get infallibility 1 automatically and do not need to check continuity in the first place. We now show $\Phi(f_\lambda)$ is continuous when Q is deterministic or partially deterministic but otherwise continuous.

Theorem 4.7. *If the random variable Q defined in Eqn. 7 is* continuous*, then $\Phi(f_\lambda)$ is continuous with respect to $\lambda \geq 0$.*

Proof. For $q \in \Delta$, define

$$\tilde{f}_\lambda(q) = \arg\max_{v \in V} (r_v + \lambda) q_v.$$

Then it follows that $\forall x \in \mathcal{X}$, $f_\lambda(x) = \tilde{f}_\lambda(\vec{p}_{Y|X}(x))$. We also define

$$\Gamma_v(\lambda) = \{q \in \Delta : (r_v + \lambda) q_v > (r_{v'} + \lambda) q_{v'}, \forall v' \neq v\}.$$

to be the open polyhedron in Δ such that $\tilde{f}_\lambda(q) = v, \forall q \in \Gamma_v(\lambda)$, *i.e.* the set of probability vector that will lead to prediction on node v given λ. Also let

$$\begin{aligned}\bar{\Gamma}(\lambda) = &\{q \in \Delta : \exists v', v'', \forall u \neq v', u \neq v'', \\ &(r_{v'} + \lambda) q_{v'} = (r_{v''} + \lambda) q_{v''} \geq (r_u + \lambda) q_u\},\end{aligned}$$

that is, the set of probability vectors that lie on the decision boundary. It is then a simple exercise to check that $\bar{\Gamma}(\lambda)$ and $\Gamma_v(\lambda), \forall v \in V$ are disjoint partitions of Δ. Then

$$
\begin{aligned}
\Phi(f_\lambda) &= \mathbb{E}_X \mathbb{E}_{Y|X}[f_\lambda(X) \in \pi(Y)] \\
&= \mathbb{E}_Q \mathbb{E}_{X|Q} \mathbb{E}_{Y|X,Q}[f_\lambda(X) \in \pi(Y)] \\
&= \mathbb{E}_Q \mathbb{E}_{X|Q} \sum_{y \in \mathcal{Y}} [f_\lambda(X) \in \pi(y)] p_{Y|X}(y|X) \\
&= \mathbb{E}_Q \mathbb{E}_{X|Q} \sum_{y \in \mathcal{Y}} [\tilde{f}_\lambda(Q) \in \pi(y)] Q_y \\
&= \mathbb{E}_Q \sum_{y \in \mathcal{Y}} [\tilde{f}_\lambda(Q) \in \pi(y)] Q_y \\
&= \int_\Delta \sum_{y \in \mathcal{Y}} [\tilde{f}_\lambda(q) \in \pi(y)] q_y \, p_Q(q) \, \mathrm{d}q \\
&= \left(\int_{\bar{\Gamma}(\lambda)} + \sum_{v \in V} \int_{\Gamma_v(\lambda)} \right) \sum_{y \in \mathcal{Y}} [\tilde{f}_\lambda(q) \in \pi(y)] q_y \, p_Q(q) \, \mathrm{d}q \\
&= \sum_{v \in V} \int [q \in \Gamma_v(\lambda)] q_v p_Q(q) \, \mathrm{d}q.
\end{aligned}
$$

Note that the first two equalities are by iterated expectations. Also we can drop $\int_{\bar{\Gamma}(\lambda)}$ at the last step because $\bar{\Gamma}(\lambda)$ has dimensions less than $|\mathcal{Y}|-1$ and therefore has zero measure.

Let $\phi_v(\lambda, q) = [q \in \Gamma_v(\lambda)] q_v p_Q(q)$. To prove continuity, it suffices to show that for each v, $\int \phi_v(\lambda, q) \, \mathrm{d}q$ is continuous with respect to λ, *i.e.* for sequences $\{\lambda_n\}$, if $\lim_{n\to\infty} \lambda_n = \lambda$, then $\lim_{n\to\infty} \int \phi_v(\lambda_n, q) \, \mathrm{d}q = \int \phi_v(\lambda, q) \, \mathrm{d}q$. This is directly implied by Lebesgue's dominated convergence theorem if we can show (1)$\lim_{n\to\infty} \phi_v(\lambda_n, q) = \phi_v(\lambda, q)$ almost everywhere and (2) for all n and every q, $|\phi_v(\lambda_n, q)| \le \psi(q)$ for some integrable ψ.

Note that condition(2) is trivial as $\phi_v(\lambda_n, q) \le p_Q(q)$ and we only need to check condition(1). First note that condition(1) trivially holds for any $q \notin \Delta$. For $q \in \Delta$, there are three possibilities: (i)$q \in \Gamma_v \lambda$, (ii)$q \in \Gamma_u(\lambda)$ for some $u \neq v$, or (iii)$q \in \bar{\Gamma}(\lambda)$. We only need to check (i) and (ii) because $\bar{\Gamma}(\lambda)$ has zero measure.

For (i), let $\gamma_v(\lambda, q) = (r_v + \lambda) q_v - \max_{v' \neq v} (r_{v'} + \lambda) q_{v'}$. For any $q \in \Gamma_v(\lambda)$, as $n \to \infty$, $\gamma_v(\lambda_n, q) \to \gamma_v(\lambda, q) > 0$. Therefore there exists n' such that $\forall n > n', \gamma_v(\lambda_n, q) > 0$ and thus $\forall n > n', [q \in \Gamma_v(\lambda_n)] = 1$. Therefore $[q \in \Gamma_v(\lambda_n)] \to 1 = [q \in \Gamma_v(\lambda)]$.

For (ii), since $q \in \Gamma_u(\lambda)$ for some $u \neq v$, as $n \to \infty$, $\gamma_v(\lambda_n, q) \to \gamma_v(\lambda, q) < 0$ and therefore $[q \in \Gamma_v(\lambda_n)] \to 0 = [q \in \Gamma_v(\lambda)]$.

□

Lemma 4.8. *If the random variable Q defined in Eqn. 7 is partially deterministic but otherwise continuous, then $\Phi(f_\lambda)$ is continuous with respect to $\lambda \ge 0$.*

Proof. When $Q \in \Delta^\ddagger$, let $y^\ddagger$ be the leaf node such that $Q_{y^\ddagger} = 1$, *i.e.* $Y = y^\ddagger$ with probability 1. Then $p_{Y|X}(v|X) = 1$ for any $v \in \pi(y^\ddagger)$ and $p_{Y|X}(v|X) = 0$ otherwise. Therefore $f_\lambda(X) = \arg\max_{v \in V} r_v p_{Y|X}(v|X) = \arg\max_{v \in \pi(y^\ddagger)} r_v$. Thus when $Q \in \Delta^\ddagger$, $f_\lambda(X) \in Y$ with probability 1. Then

$$
\begin{aligned}
\Phi(f_\lambda) =& \mathbb{E}_{X,Y|Q \in \Delta^\ddagger}[f_\lambda(X) \in \pi(Y)] \Pr(Q \in \Delta^\ddagger) \\
&+ \mathbb{E}_{X,Y|Q \notin \Delta^\ddagger}[f_\lambda(X) \in \pi(Y)] \Pr(Q \notin \Delta^\ddagger) \\
=& \Pr(Q \in \Delta^\ddagger) \\
&+ \mathbb{E}_{X,Y|Q \notin \Delta^\ddagger}[f_\lambda(X) \in \pi(Y)] \Pr(Q \notin \Delta^\ddagger).
\end{aligned}
$$

The first term is constant with respect to λ so we only need to show the continuity of the second term, which can be proved the same way as in Theorem 4.7 by replacing p_Q with $p_{Q|Q \notin \Delta^\ddagger}$. □

4.5. Complete algorithm

We have shown that under very realistic conditions, strong duality holds and we can achieve the optimal solution. Now are are ready to present the complete algorithm(Algorithm 1). It can be understood very intuitively.

Algorithm 1 Solving Infallible classification(OP1)

1. Solve OP2 and obtain a solution f_0.
2. If $\Phi(f) \ge 1 - \epsilon$, return f.
3. Binary search to find a $\lambda \in (0, \bar{\lambda}]$. For each λ, obtain f_λ by solving OP2 with transformed rewards $\hat{r}_v = r_v + \lambda, \forall v \in V$. Stop when $\Phi(f_\lambda)$ is close enough to $1 - \epsilon$ or when the maximum number of iterations is reached. Return the last f_λ.

We first discard the constraint and see if we can satisfy infallibility. If yes, then we are done. If not, we increase the reward of each node by $\lambda > 0$ such that the root reward gets relatively bigger, which encourages predictions to go to the root and make the infallibility increase. We try to find a λ that matches the infallibility closely by binary search.

4.6. Calibration

We have been assuming that we can obtain Bayesian optimal classifiers, which is impossible in most cases. What if we can only learn classifiers far from Bayesian optimal?

The answer is yes. We can make any pre-trained multiclass classifier nearly infallible, even if they perform very poorly, provided that we can calibrate them well. In this section, we first introduce the concept of calibration and show that the same algorithm can be applied.

Let a classifier $g : \mathcal{X} \to \Delta$ map an example $x \in$ to multinomial probabilities on leaf nodes. Let $G = g(X)$ and G is also a random variable. A classifier g is *calibrated* if $\Pr(Y = y|G = q) = q_y, \forall q \in \Delta, \forall y \in \mathcal{Y}$, *i.e.* if we take

all examples that are predicted leaf probabilities q, then the percentage of examples that have ground truth label y is q_y for any leaf node y. Note that a Bayesian optimal classifier is by default calibrated, but a calibrated classifier is not necessarily Bayesian optimal. It merely state its confidence truthfully. For example, in binary classification, assuming that the two classes are evenly distributed, a classifier that always outputs 50% probability is perfectly calibrated yet useless.

We can make a pre-trained, calibrated classifier g nearly infallible by learning a function $h : \Delta \to V$ that takes the output of g and predict a node. Then we have a similar optimization problem.

OP1'. Nearly infallible classification for pre-trained calibrated classifiers.

$$\begin{aligned} \underset{h}{\text{maximize}} \quad & R(h \circ g) \\ \text{subject to} \quad & \Phi(h \circ g) \geq 1 - \epsilon. \end{aligned}$$

In fact, it is easy to show that Algorithm 1 still applies with a simple change that

$$f_\lambda(x) = h \circ g(x) = \arg\max_{v \in V} (r_v + \lambda) g_v(x).$$

Similar to the conditions for Q defined in Eqn. 7, Algorithm 1 is optimal for OP1' if $G = g(X)$ is deterministic, continuous or partially deterministic but otherwise continuous. All proofs trivially carry over if we replace $p_{Y|X}(v|x)$ with $g_v(x)$ and use the following identities:

$$\begin{aligned} &\Phi(h \circ g) = \mathbb{E}\left[h(g(X)) = \pi(Y)\right] \\ &= \mathbb{E}_G \mathbb{E}_{Y,X|G}\left[h(G) = \pi(Y)\right] = \mathbb{E}_G \mathbb{E}_{Y|G}\left[h(G) = \pi(Y)\right], \end{aligned}$$

and similarly $R(h \circ g) = \mathbb{E}_G \mathbb{E}_{Y|G}\, r_{h(G)}[h(G) = \pi(Y)]$. We provide complete proofs for the pre-trained case in supplemental materials.

A remaining question is how to obtain calibrated classifiers. In principle any classifier can be calibrated as long as they output some confidence scores. The next step is to map them to calibrated probabilities. Most classifiers give direct probability estimates, either explicitly such as logistic regression, or implicitly such as boosting. SVM does not give probability estimation but can be post-processed to do so, *e.g.* by fitting a logistic function [16]. In any case, even with direct probability output, additional calibration is necessary because the raw output can be biased. For all models, we can use isotonic regression, a general technique proposed in [**?**] for calibration.

5. Experiments

We use three datasets,ImageNet65, ImageNet1K and ImageNet10K. ImageNet65 is a simplified four level tree from ImageNet with 57 leaf nodes and 8 internal nodes. The structure is shown in Fig. 5. We sample 100 images per category from the ImageNet database [8] as training, 50 as validation and 100 as test. ImageNet1K is the same dataset used in ILSVRC2010 [**?**] and we use the same training, validation and test splits. ImageNet10K is 9 million images from 10184 categories used in [**?**]. We use 50% as training, 25% as validation and 25% as test.

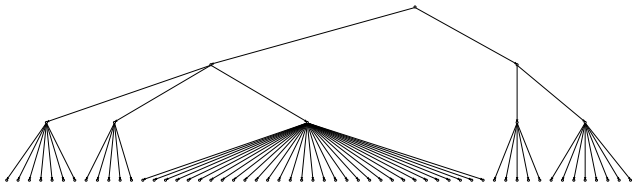

Figure 2. The tree structure of ImageNet65. The tree is designed to be not well balanced to approximate the real world hierarchy such as WordNet.

We evaluate three variants of our proposed primal dual algorithm and one baseline as an extension from [18]

- LR+PD. Train one-versus-all logistic regression classifiers on leaf nodes and obtain posterior probabilities. Posterior probabilities on internal nodes are summation of those of leaf nodes. We then use the primal dual algorithm to find the optimal decision rule.
- SVM+PD. Similar to LR+PD except that we use one versus all SVM and calibrate with Platt's scaling [**?**].
- Tree+PD. Similar to LR+PD except that we learn a decision tree by training one-versus-all logistic regression classifiers at each node. The posterior probabilities on internal nodes are obtained from the internal classifiers.
- GL+Th. Learn a decision tree model similar to [18]. An example will stay at an internal node if it's posterior probability is below a threshold. Use binary search to find a global threshold closest to the infallibility constraint while satisfying it.

We set our infallibility requirement $1-\epsilon$ to various values between 0.3 and 0.99 and plot the information gain versus the actual infallibility on for all methods in Fig. 5. We show that all curves produced by our primal dual algorithm beat

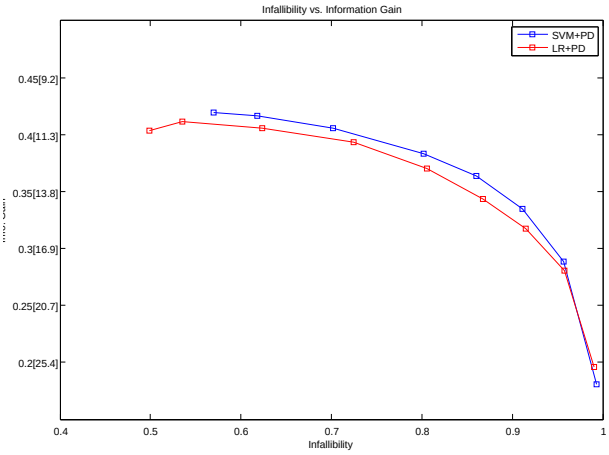

Figure 3. Information gain versus infallibility on ImageNet65. The numbers in brackets on Y axis indicates the equivalent of number of uncertain categories.

the naive global threshold method regardless of the learning method used.

We then demonstrate how our algorithm produces non-trivial, useful infallible classifiers by showing the distribution of predictions over the semantic levels and their respective accuracy in Fig. [**?**].

We also plot the infallibility versus the number of the iterations for the binary search in Algorithm 1. We show that it converges very quick to optimal.

We next show results on ImageNet1K and ImageNet10K. They are awesome.

We finally show example images that predicted to different levels on ImageNet1K in Fig. 5

6. Conclusion and Future work

We propose and solved the problem. Future work would be to extend it to detection, image parsing and other more complex tasks and explore the weakly supervised settings.

References

[1] Y. Amit, M. Fink, and N. Srebro. Uncovering shared structures in multiclass classification. In *In Proceedings of the*

figures/dist_level.pdf

Figure 4. A bar plot of distribution of predictions made by the SVM+PD infallible classifier with a 90% infallibility guarantee.

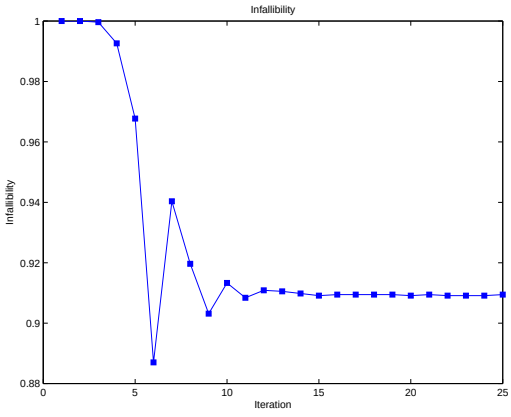

Figure 5. Infallibility versus number of iterations with 90% guarantee in the binary search of Algorithm 1.

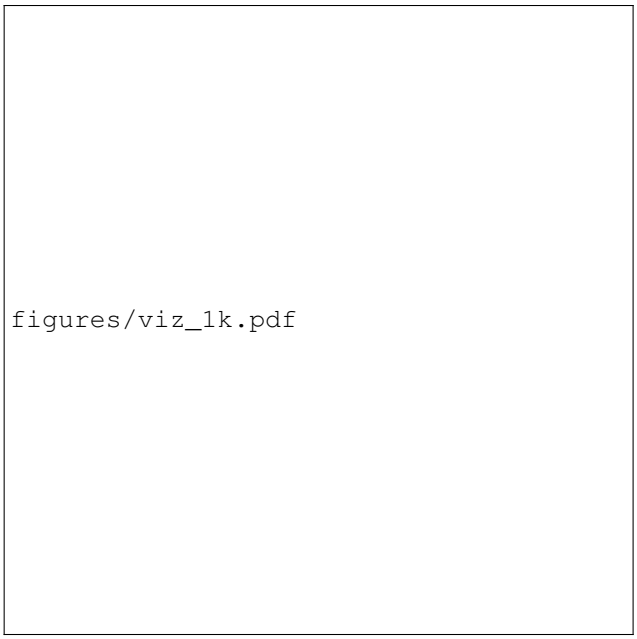

Figure 6. Example images predicted to different semantic levels for ImageNet1K.

ing, pages 17–24. Springer-Verlag, 2007. 2

[2] P. L. Bartlett and M. H. Wegkamp. Classification with a reject option using a hinge loss. Technical report, U.C. Berkeley, 2006. 3

[3] S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. In *Advances in Neural Information Processing Systems (NIPS)*, 2010. 2

[4] S. S. Bucak, R. Jin, and A. K. Jain. Multi-label learning with incomplete class assignments. In *CVPR*, pages 2801–2808, 2011. 3

[5] N. Cesa-bianchi, L. Zaniboni, and M. Collins. Incremental algorithms for hierarchical classification. In *Journal of Machine Learning Research*, pages 31–54. MIT Press, 2004. 2

[6] C. Chow. On optimum recognition error and reject tradeoffs. 16(1):41–46, January 1970. 3

[7] T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *JMLR*, 2011. 3

[8] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR09*, 2009. 7

[9] H. Everett. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11(3):399–417, 1963. 4

[10] E. Grall-Maes and P. Beauseroy. Optimal decision rule with class-selective rejection and performance constraints. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(11):2073 –2082, nov. 2009. 3

[11] T. Ha. The optimum class-selective rejection rule. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(6):608 –615, jun 1997. 3

[12] J. José del Coz and A. Bahamonde. Learning nondeterministic classifiers. *J. Mach. Learn. Res.*, 10:2273–2293, December 2009. 3

[13] M. Marszalek and C. Schmid. Semantic hierarchies for visual object recognition. In *CVPR07*, pages 1–7, 2007. 2

[14] H. Masnadi-shirazi and N. Vasconcelos. Risk minimization, probability elicitation, and cost-sensitive svms. In *ICML*, 2010. 2

[15] H. Masnadi-Shirazi and N. Vasconcelos. Cost-sensitive boosting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(2):294–309, 2011. 2

[16] J. Platt. Probabilistic outputs for support vector machines and comparison to regularize likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74, 2000. 7

[17] A.-M. Tousch, S. Herbin, and J.-Y. Audibert. Semantic hierarchies for image annotation: A survey. *Pattern Recogn.*, 45:333–345, January 2012. 2

[18] A. Vailaya, M. A. T. Figueiredo, A. K. Jain, and H. Zhang. Content-based hierarchical classification of vacation images. In *ICMCS, Vol. 1*, pages 518–523, 1999. 2, 7

[19] M. Yuan and M. Wegkamp. Classification methods with reject option based on convex risk minimization. *J. Mach. Learn. Res.*, 11:111–130, March 2010. 3

[20] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *KDD '02*, pages 694–699, New York, NY, USA, 2002. ACM. 2

[21] A. Zweig and D. Weinshall. Exploiting object hierarchy: Combining models from different category levels. In *ICCV07*, pages 1–8, 2007. 2